Top 20 User Interface Mistakes In Today's Apps

Chris Brown
christob@microsoft.com
Hillel Cooperman
hillelc@microsoft.com

Lead Program Managers
Windows User Interface Team

Simplicity & the Bottom Line

- Improve your app’s user-experience
- Hold on to existing customers
  - Review top customer complaints
  - How to fix/avoid them
- Expand your reach
  - Review overlooked customer scenarios
  - How to easily support them

Windows® Apps Today

- Highlights
  - Tons of available Windows apps
  - Creativity and quality is always improving
  - Healthy re-use of shared components: Web platform, controls, multimedia
- Challenges - Simplicity, simplicity,

  simplicity!
  - Customer satisfaction with the PC is shaky
  - Support costs are high

The Simplicity Challenge

- Simplicity is fundamental to your success
  - “PCs have too many features”
  - “I can’t find the features I care about”
  - “I can’t figure out how to use the features I do find”
  - Simplicity requires **effort**
- Your app is just one piece of the PC
  - Compete on the shelf, not on the screen!
  - Customers view PC as a single appliance
  - We all own the Windows platform
- Microsoft is far from perfect
  - Be critical, give us your feedback

Agenda

- Top 20 UI mistakes in Windows apps today
- For each mistake:
  - Problem, Solutions/strategies, Benefits
- Areas
  - Install/uninstall
  - Using the file system
  - GUI overload
  - Neglected scenarios

Install/Uninstall

- Don’t wait till the end of your development schedule!
  - Setup is the first thing customers see
  - Uninstall is crucial for system robustness
- Overview
  - Mistake #1: Weird setup UI
  - Mistake #2: Awkward first experience
  - Mistake #3: Uninstall hell

Mistake #1 - Weird setup UI

Problems

- Too many steps, too many clicks
- The three “R”s of bad setup UI:
  - Reboots, READMEs, and random windows

Strategies/solutions

- Simplify default setup experience
  - Typical versus custom install
  - Minimize legal disclaimers
- Avoid reboots
  - Use Windows Installer
  - Search for running DLLs, prompt or offer to close running apps

Mistake #1

Weird setup UI

Strategies/solutions (cont.)

- Don’t over-expose READMEs
  - Think ahead!
  - Crucial support info: Add a link from Help menu
  - Otherwise, leave it in your app folder, or on the CD
- Avoid random windows
  - Consolidate progress dialogs
  - Use IShellLink to create Start Menu icons, not DDE

Benefits

- Positive first experience

Mistake #2

Awkward first experience

Problem: “I finished Setup, now what?”

- I want to run the app right away
- I want to perform top tasks

Solutions/strategies

- Auto-launch after setup
  - Offer to run the program immediately
  - Remind user how to launch program in the future (e.g., name of start menu icon)
- Immediate productivity
  - Tours help, but aren’t enough
  - Initial screen should clearly list top *n* tasks

Mistake #2

Awkward first experience

Solutions/strategies (cont.)

- Avoid feature-overload
  - Slow-reveal features over time: Tips, intelligent help

Benefits of a good first experience

- First impressions last a long time
- Great for product evaluators/influential end-users

Mistake #3
Uninstall hell

Problems:

- Missing uninstaller
- CD-ROM required to start the uninstall
- Failed/incomplete uninstall

Solution:

- Use Windows Installer!
- Commitment from top setup toolkits,
  e.g., InstallShield

Mistake #3

Uninstall hell

- Resources
  - MGMT03/04 - Manageable Setups, Parts I and II
  - MGMT05 - Dynamic/Robust Applications
  - http://www.microsoft.com/management
- Benefits:
  - Great uninstall, self-repairing, install-on-demand, integrated with Windows NT® 5.0 Control Panel

Using The Filesystem

- “My hard disk is a mess”
  - Documents are hard to find (esp. in corps)
  - Files on my hard disk are random, cryptic, and out of control
  - Uninstallers and disk cleaners are top-selling Windows apps

Using The Filesystem

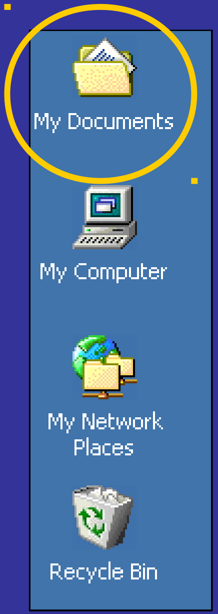
- Overview
  - Mistake #4: No support for “My Documents”
  - Mistake #5: Improper placement of program files
  - Mistake #6: Improper placement of user docs and settings
  - Mistake #7: Reckless temporary storage
  - Mistake #8: Cryptic files

Mistake #4

Missing support for “My Documents”

Problem: “Where did my documents go?”

- Bad: Storing docs under Program Files” or app folder
- Bad: Hard-coding “C:\My Documents”
- Bad: Creating your own doc folder outside of “My Documents”



Solutions/strategies:

- Save user-created documents directly to My Documents
  - Use Shell APIs to locate the folder

```
#include <shlobj.h>
// this API is in shfolder.dll for non-NT5 platforms
SHGetFolderPath(NULL, CSIDL_PERSONAL, 0, NULL, szPath);
```

Mistake #4

Missing support for “My Documents”

Solutions/strategies (cont):

- Create folders under My Documents for large projects (e.g., VB projects, Web authoring)
- Leverage “My Pictures” for acquired Images (cameras, scanners)
- See slides from “Leveraging Windows 98 and Windows NT 5.0”

“My Pictures”: See samples on your DVD for more details

Benefits of My Documents

- Easy access from desktop, file-open
- Auto support for per-user, roaming, remote/secure storage
- Auto support for different OS languages
- Auto support for renaming “My Documents”

Mistake #5

Improper placement of program files

Problem: “Where did these files come from?”

- Bad: Creating random files at root of hard disk
- Bad: Creating app folders at root of hard disk
- Bad: Creating files to Windows folder (outside profiles)
- Bad: Hard-coding “C:\Program Files”
- Bad: Confusing/cluttered app folders

Solutions/strategies:

- Create a single app folder under Program Files
  - How to locate “Program Files”

```
#include <shlobj.h>
// this API is in shfolder.dll for non-NT5 platforms
SHGetFolderPath(NULL, CSIDL_PROGRAM_FILES, 0, NULL, szPath);
```

Mistake #5

Improper placement of program files

Solutions/strategies (cont.):

- Save read-only program files in your app folder
  - Rule of thumb: Would your uninstaller delete the file?
    - Yes-> it belongs under Program Files
    - No-> it does not belong under Program Files (See mistake #3)
  - Examples
    - Good: Clip-art, samples, libraries (primarily read-only)
    - Bad: Edited templates, custom dictionaries (See mistake #3)
- Clean up your app folder
  - Bury cryptic files, or mark as hidden
  - Consider using Web view: hide app files, present links to uninstaller/support

Mistake #5

Improper placement of program files

Solutions/strategies (cont.):

- Use “Program Files\Common Files” for shared components

  ```
  HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion,
  CommonFilesDir
  ```

  - Use registry look-up to locate the proper folder
  - Register per-app paths instead of using \Windows\System

Benefits of “Program Files” - Robustness!

- Separation of data from implementation goo
- “Soft-barrier” protects your naïve users from screw ups

Mistake #6

Improper placement of application data

Some definitions:

- Program files = Read-only (implementation files, libraries)
- My Documents = User-visible files (“file-save”)
- Application data = everything else

Mistake #6

Improper placement of application data

Examples:

- Custom dictionaries
- E-mail store
- Internet cache
- Offline mail storage
- User history
- Log files

Problems: App data is all over the hard disk

- Files not clearly marked, not clearly organized
- Broken scenarios: Backup, roaming,

Mistake #6

Improper placement of application data

Solutions/strategies

- Use the three system-defined Application Data folders

  1) *username*\Application Data == Roaming, per-user

  2) *username*\Application Data\Local Settings == Local, per-user

  3) All Users\Application Data == Local, common

Mistake #6

Improper placement of application data

- Decision #1: To roam or not to roam
  - Roaming requires tree-copy on every logon and logoff
  - Do roam: Important data, user-customizations
  - Don’t roam: Discardable data, large temporary files

Mistake #6

Improper placement of application data

- Decision #2: Per-user versus all-users
  - Avoid “all-user” app data
    (no roaming, hard to access)
  - Only use “all-user” app data folder as a last resort
  - Examples: Log files, hidden databases and index files

Mistake #6

Improper placement of application data

Solutions/strategies (cont.)

- How to locate the three app data folders
- Benefits of “application data”

```
#include <shlobj.h>
// this API is in shfolder.dll for non-NT5 platforms
SHGetFolderPath(NULL, CSIDL_APPDATA, 0, NULL, szPath);
SHGetFolderPath(NULL, CSIDL_LOCAL_APPDATA, 0, NULL, szPath);
SHGetFolderPath(NULL, CSIDL_COMMON_APPDATA, 0, NULL, szPath);
```

- SHFOLDER.DLL lets you support all win32 platforms
- Easier for users/admins to backup the right stuff
- Automatic roaming, profiles, administrative scenarios; language independent

Summary - Where to put your files

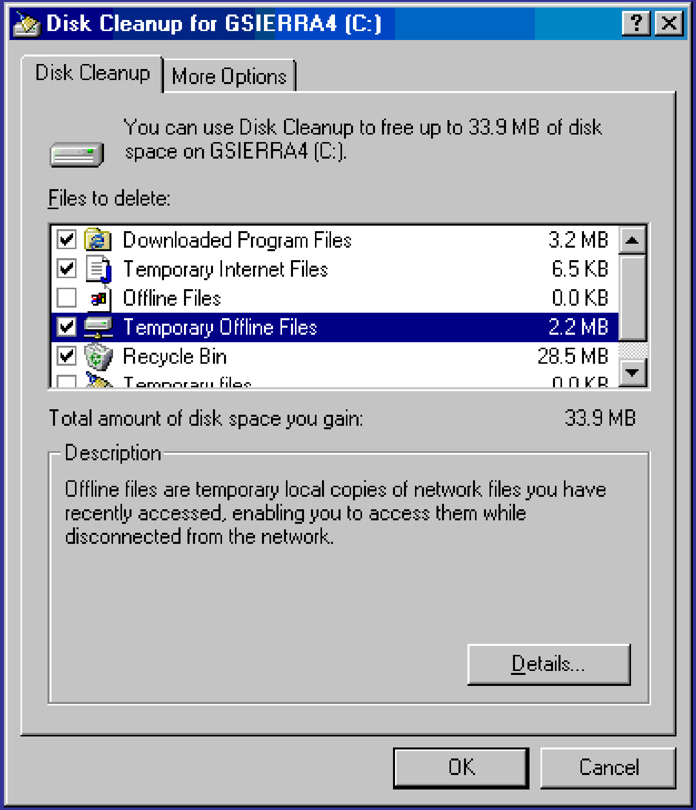
- “My Documents”
  - User-created files
- “Program Files\appname”
  - Read-only implementation files and libraries/samples
- “Profiles\\username\\Application Data\appname”
  - Auto-created user files (e.g., e-mail DB, custom dictionaries)
- “Profiles\\username\\Application Data\\Local Settings\appname”
  - Discardable/local-only user data files (e.g., offline stores)

- “My Documents\\My Pictures”
  - Acquired images (cameras/scanners)
- “Profiles\\All Users\\Application Data\appname”
  - Last-resort for local-only dynamic data (e.g., log files)
- “Program Files\\Common Files\companyname”
  - For shared services/components (e.g., DAO libraries)

Mistake #7
Reckless temporary storage

Problem: Disk-caches all over the place

- The Windows NT 5.0 worst-case:
  - Recycle bin
  - Internet Explorer: Temporary Internet Files
  - Offline Files cache
  - Outlook Express offline storage
  - Index Server: Content indexes



You can use Disk Cleanup to free up to 33.9 MB of disk space on GSIERRA4 (C:).

Files to delete:

☑	Downloaded Program Files	3.2 MB
☑	Temporary Internet Files	6.5 KB
□	Offline Files	0.0 KB
☑	Temporary Offline Files	2.2 MB
☑	Recycle Bin	28.5 MB
□	Temporary files	0.0 KB

Total amount of disk space you gain: 33.9 MB

Description

Offline files are temporary local copies of network files you have recently accessed, enabling you to access them while disconnected from the network.

Details in "Disk Cleaner Plugins" Document on DVD

Solution: Disk Cleaner to the rescue!

- One-stop extensible cleanup UI
- Access points: Low disk-space warning,

Mistake #8
Cryptic files

Problem: Mysterious files of unknown origin

- Examples: ‘exchange.ost’, ‘catalog.wci’, ‘ffastun.ffo’, ‘suhdlog.dat’
- Like wires sticking out of your stereo

Solutions/strategies:

- Super-hidden files
  - On Windows NT 5.0: Files marked with both system + hidden bits are never shown in explorer
- Use File-types to ensure accountability
  - Every file you create should have a registered file-type,
    no exceptions
  - Create new file types even for implementation/data files
    - Friendly type description e.g., “Outlook Offline Mail Files”
    - Friendly icon

Mistake #8
Cryptic files

Solutions/strategies (cont.):

- Use long file names!
  - It’s your code, and LFN support is little to no work
  - Multi-language products: Consider using a registry key to point to your files

Benefits

- Clear file descriptions and accountability
- Users know what to back up, what to discard

GUI Overload

- "My screen looks like Times Square"
  - Competing UI = Customer confusion, lower productivity, lower satisfaction, numbness
  - Focus should be on top user tasks, not on feature advertisement
- Top mistakes:
  - Mistake #9: Start Menu abuse
  - Mistake #10: Desktop icon abuse
  - Mistake #11: App UI at startup time
  - Mistake #12: Taskbar status abuse
  - Mistake #13: Error messages from hell

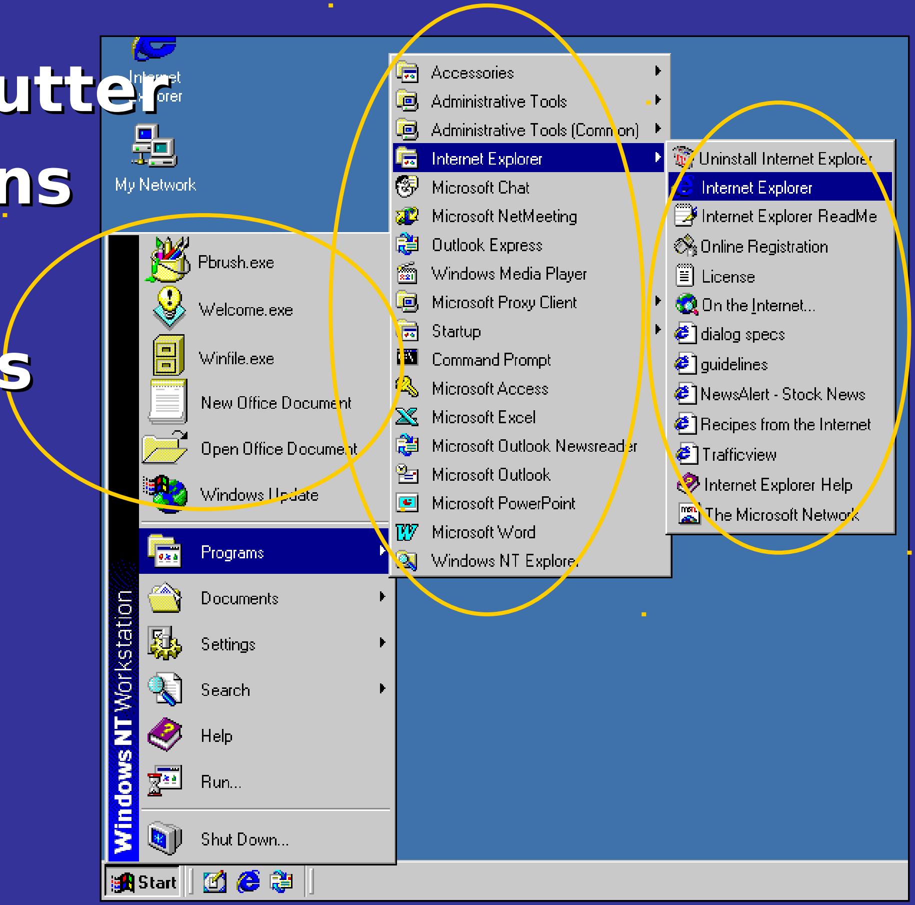
Mistake #9
Start Menu abuse

Problems: Start Menu clutter

- Too many program icons
- Too much "cascading"
- Too many useless icons

Usability news flash!

- Fewer cascades = higher click rate!
- "Uninstall" icon is unnecessary!

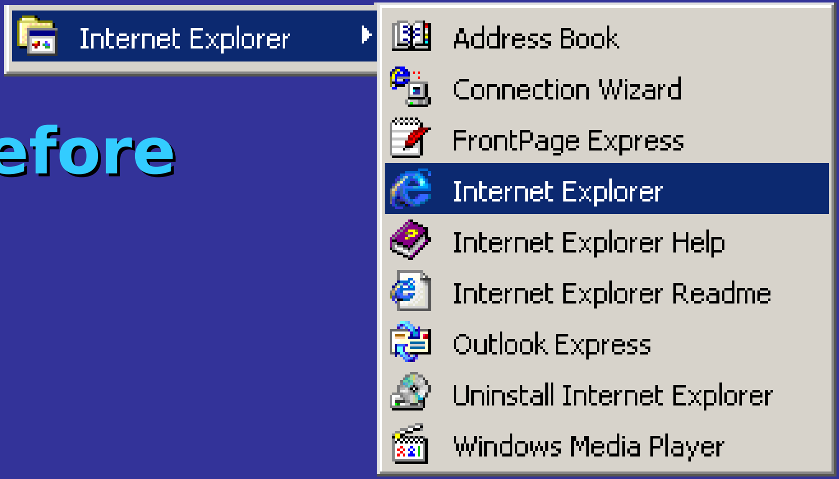


Mistake #9
Start Menu abuse (2)

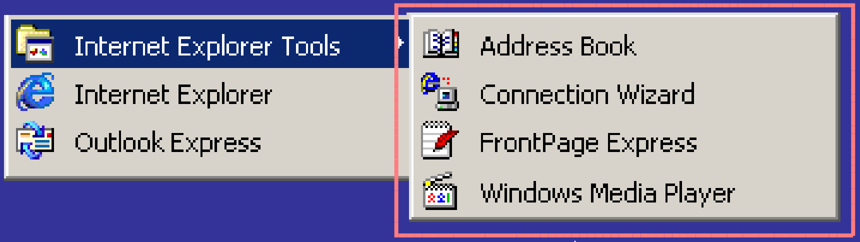
Solution/strategies:

- One icon under Start->Programs
- Kill useless icons
  - ~~HELP~~
    ~~README~~
    ~~UNINSTALL~~
- Flatten app-suites
- If you must, create a single group, only for utilities
- Avoid the top of the Start Menu

Before



After



↑

Avoid group

if possible

Benefits:

- Higher click rates, faster launching
- Less clutter amongst your OWN app icons
- Focus on top tasks, not install/support

Mistake #10
Desktop icon abuse

Problem: Too many desktop icons!

Usability news flash!

- Negative customer reaction “The desktop is my space!”
- Desktop is hard to get back to

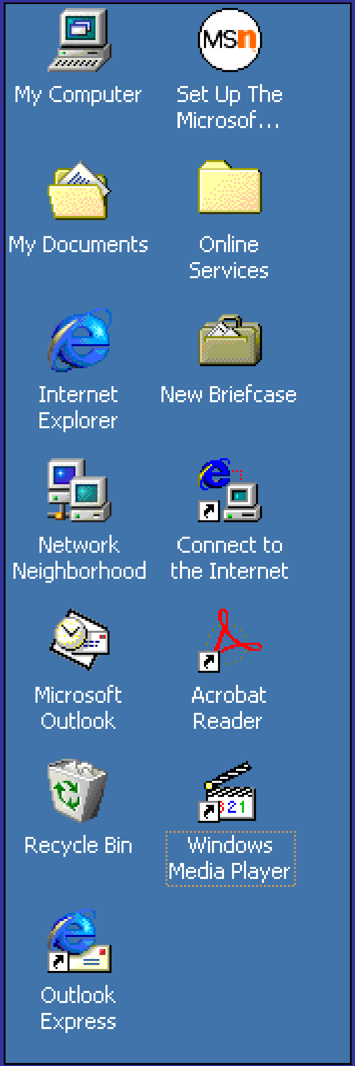
Solution/strategies:

- Use the Start Menu
- Fair exceptions: E-mail, Web
- Ask before adding icons

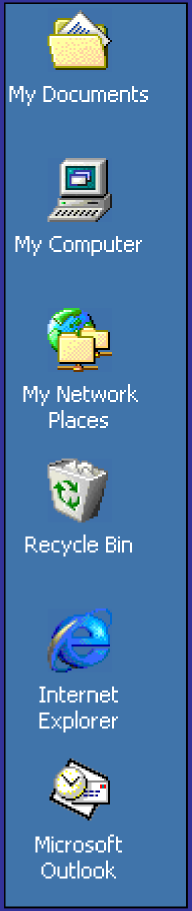
Benefits

- Avoid customer turn-off
- Optional/custom desktop icons are a service, not a nuisance

Bad



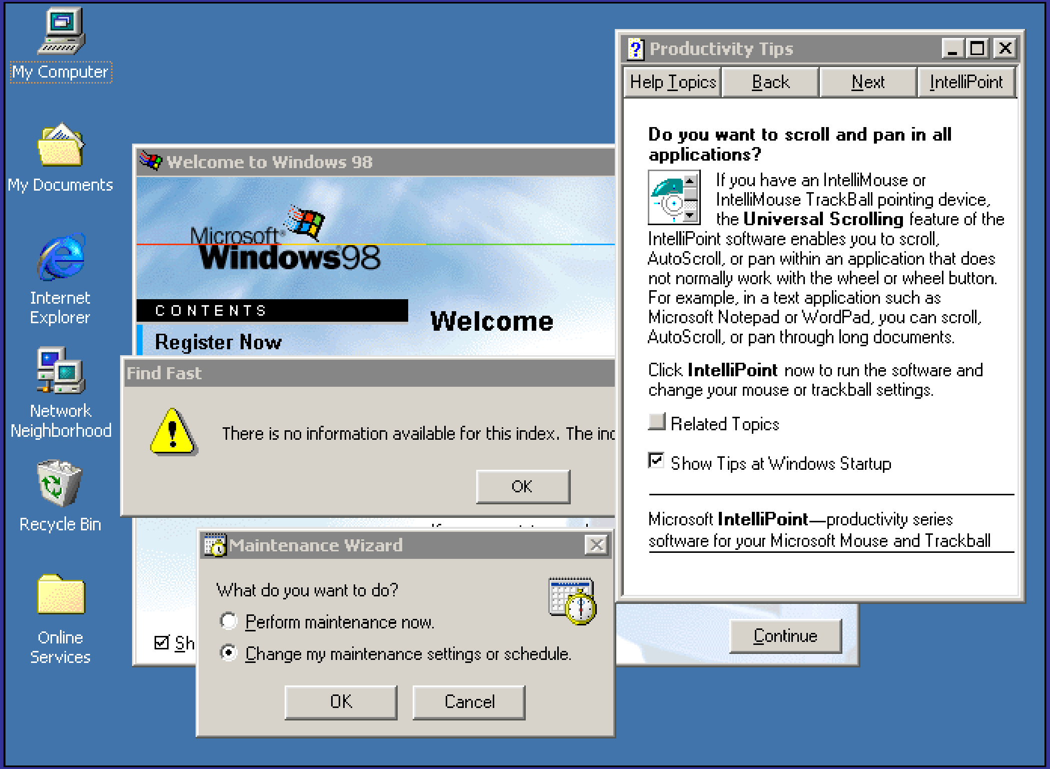
Good



Mistake #11
App UI at startup time

Problem: Dialog-overload at session start

- Multiple-dialog pile-up is confusing/scary
- Lack of context makes dialogs difficult to comprehend
- “I just close all those windows anyway”



There is no information available for this index. The inc

What do you want to do?

○ Perform maintenance now.

● Change my maintenance settings or schedule.

Do you want to scroll and pan in all applications?

If you have an IntelliMouse or IntelliMouse TrackBall pointing device, the **Universal Scrolling** feature of the IntelliPoint software enables you to scroll, AutoScroll, or pan within an application that does not normally work with the wheel or wheel button. For example, in a text application such as Microsoft Notepad or WordPad, you can scroll, AutoScroll, or pan through long documents.

Click **IntelliPoint** now to run the software and change your mouse or trackball settings.

□ Related Topics

☑ Show Tips at Windows Startup

Microsoft **IntelliPoint**—productivity series software for your Microsoft Mouse and Trackball

Mistake #11

App UI at startup time

Solution: Avoid dialogs on system startup!

- Strategies
  - First-time config=> Move to Setup UI
  - Product intro dialogs=> Wait till app is launched
  - Background ops => Run silently, fail gracefully
  - Present dialogs ONLY in response to user actions
  - Kill "progress" dialogs and error messages (Mistake #10)
  - Serialize! On Windows NT 5.0, put your dialog "in the queue"

```
HKLM\Software\Microsoft\Windows\CurrentVersion\Run
Create sub-keys under the 'Run' key
Each sub-key can contain lists of Executables
Sub-keys are executed synchronously, in alphabetical order
```

- Benefits
  - More 'eyeballs' on your UI

Mistake #12
Taskbar status abuse

Problem: Taskbar clutter

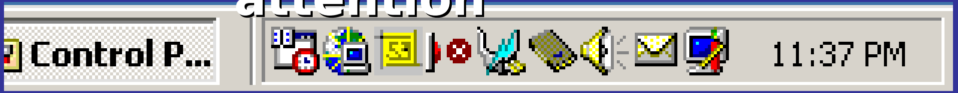
- User numbness: Most of these icons are never touched
- User annoyance: Each icon permanently eats up 16x16 pixels

Solution/strategies

- Right-side of Taskbar is for notification, not for running status

Usability news flash!

- The human eye is designed to detect motion
- Hiding/showing icons or flashing - Gets more attention



Mistake #12

Taskbar status abuse

Solution/strategies (cont.):

- For notification (A system-wide event):
  - If you have an app window already, flash its Task button
  - If you don't have an app window running, create a temporary icon that disappears after a short delay
  - Consider using balloon tips (see next slide)
- For status:
  - Use your app window if possible
  - Extend existing Control Panels (Keyboard, mouse, display, etc.)
  - Status-only Taskbar icons should be optional, off by default

Benefits:

- Motion gets more attention
- Avoid customer numbness, annoyance

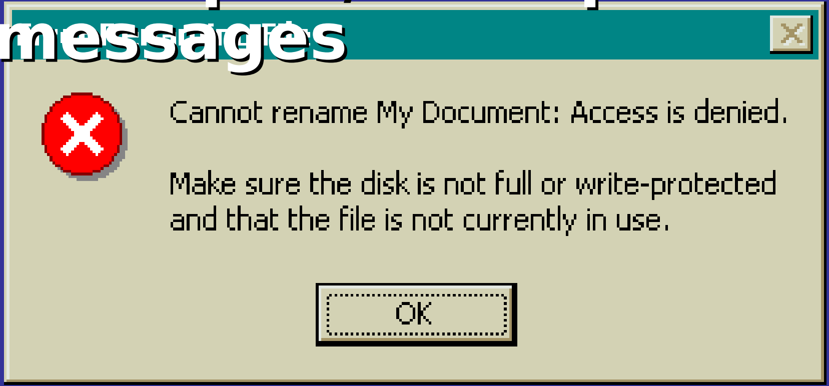
Survive explorer.exe crashes!   See the DVD for more details

Mistake #13

Error messages from hell

Problems

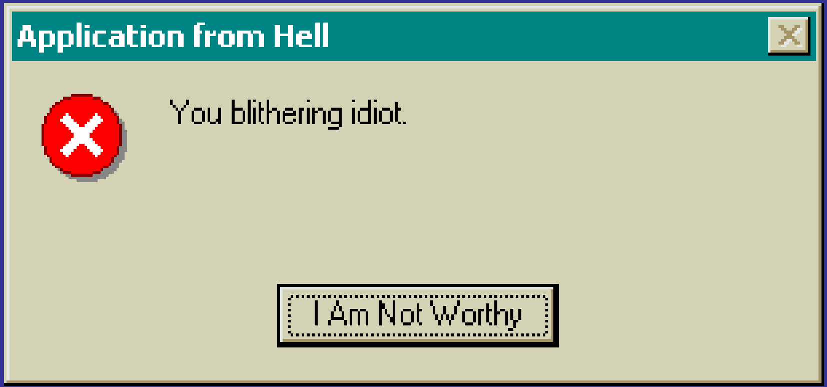
- Unhelpful/incomprehensible error messages



Cannot rename My Document: Access is denied.

Make sure the disk is not full or write-protected and that the file is not currently in use.

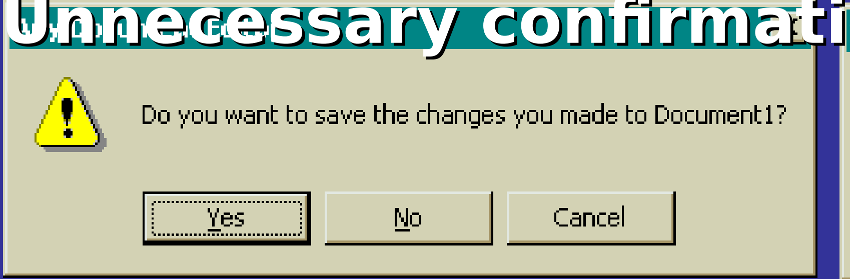
Example



You blithering idiot.

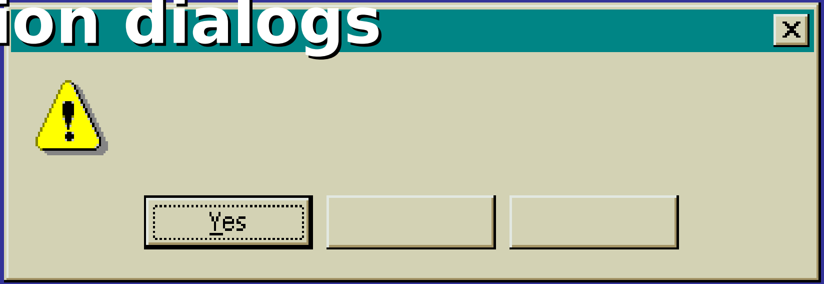
What the user sees

- Unnecessary confirmation dialogs



Do you want to save the changes you made to Document1?

Example



What the user sees

Mistake #13

Error messages from hell

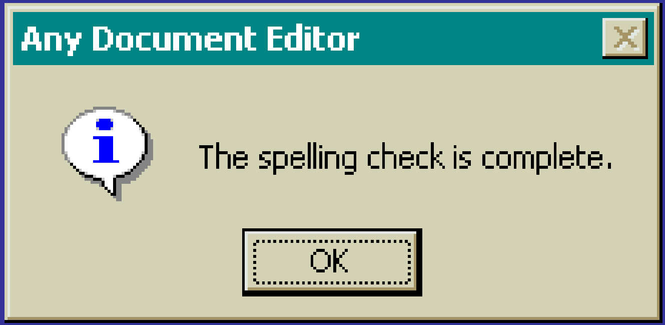
Problems (cont.)

- Unnecessary alerts

Solutions/strategies

- Avoid dialogs in the first place -
  Disable controls, provide advance
  feedback, cope with any input,
  fail gracefully
- If you must - Use normal language,
  don't blame the user,
  always provide an action path
- Don't steal focus;
  consider balloon tips

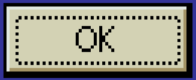
See the DVD for more details



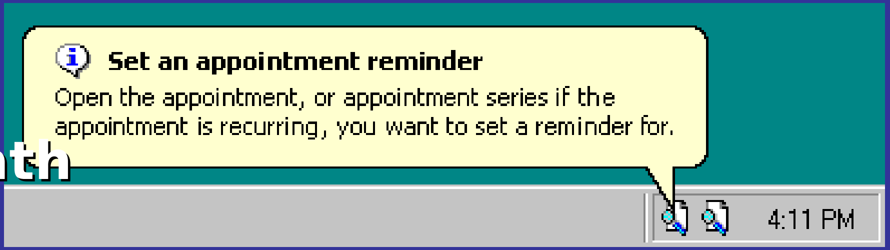
Any Document Editor

The spelling check is complete.

Example



What the user sees



Set an appointment reminder

Open the appointment, or appointment series if the appointment is recurring, you want to set a reminder for.

Neglected Scenarios

Which customers did I forget?

- Support more customer scenarios with only a little extra effort
- Accessibility/consistency
  - Mistake #14: Slacking on keyboard UI
  - Mistake #15: Dialog conventions
  - Mistake #16: Ignoring per-user colors and fonts
- Using the display
  - Mistake #17: Forgetting about color
  - Mistake #18: Neglecting multi-monitor
- Worldwide customers
  - Mistake #19: Overlooking the system locale
  - Mistake #20: Missing support for

Mistake #14
Slacking on keyboard UI

- Who cares about the keyboard?
  - Laptop users, non-sighted users, RSI victims
- Simple things
  - Web pages
    - Use anchor tags (“javascript:”) instead of click events
    - Internet Explorer 5: Use keyboard shortcuts for form controls
  - Win32 UI
    - Every function should have a keyboard equivalent
    - Take 20 seconds to fix your TAB order
    - Include keyboard shortcuts for all your controls
- Windows NT 5.0 opportunities
  - New keyboard support: Taskbar status area,

Mistake #15

Dialog conventions

- Why does consistency matter?
  - Most customers perform tasks by rote
  - Muscle memory is important for user comfort
  - Aesthetics are important for readability (compare your screens to a printed page!)
- Simple rules to follow
  - Standard buttons (OK/cancel, yes/no)
    - Alignment: One of horizontal on bottom right or vertical on top-right
    - Order - OK, cancel, followed by other buttons
    - Keyboard shortcuts are unnecessary: OK, cancel
  - Readability conventions
    - Labels and edit boxes: Align text baselines, use colons after labels (Cues for screen-readers)
    - Use consistent margins - Seven dlus recommended

Mistake #16

Ignoring user colors and fonts

- Why are user preferences important?
  - Accessibility - High-contrast
  - Readability - All displays are different (large screens, faded laptop displays)
  - Customer satisfaction - It’s my environment!
- Simple things
  - Leverage user-chosen fonts
    - Menu, icon, tooltip, caption, palette title
    - SystemParametersInfo(): SPI_GETNONCLIENTMETRICS, SPI_GETICONTITLELOGFONT
  - Color choices: Hard-coded colors versus system colors?
    - Doesn’t have to be one or the other
    - Create minor variations based on user-chosen colors
    - Support high-contrast flag!

Mistake #17

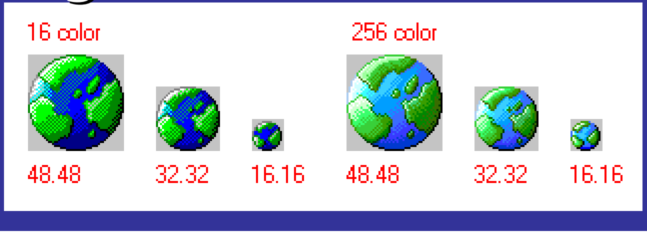
Misuse of color

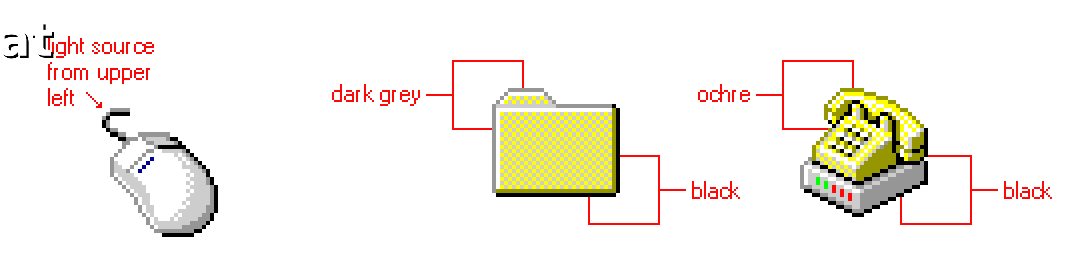
Problems

- Palette awareness, palette conflicts
- Icon conventions

Strategies/solutions

- On 8-bit displays: Use standard halftone palette
  - Includes variety of grays, 6x6x6 color cube
  - Avoids ‘palette flashes’ (used by Internet Explorer, Office)
- Respond to dynamic color-depth changes
  - WM_DISPLAYCHANGE
- Follow good icon conventions
  - Color depths/sizes
  - Border styles
  - Toolbar glyphs are flat
  - Icons are 3D

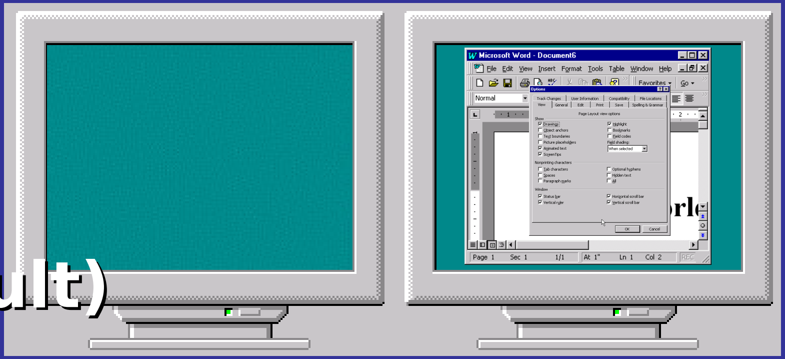




Mistake #18
Neglecting multi-monitor

Problem

- Dialogs appear on the wrong monitor
(primary monitor by default)



Solution/strategies

- Multi-monitor desktop is just one big coordinate space
- Launch secondary windows directly on top of primary windows
- Test your software

Details in “Mutltimon Support” document on the DVD

Mistake #19

Overlooking the system locale

- Worldwide support is crucial for your business
  - Even English-only apps are sold worldwide
- Some simple steps
  - Date/time formats
    - Use Datepicker control for input
    - GetDateFormat(), GetTimeFormat() for simple conversions
  - Currency, numbers
    - GetCurrencyFormat(), GetNumberFormat()
  - Sorting order
    - LCMapString() handles diacritic marks for you
- Other opportunities
  - Support separate locales for document, your

Mistake #20

Missing support for character sets

The global market

- Majority of software sales are outside North America
- Mixed language environments are extremely common

Some simple steps

- Multilingual text on a budget
  - Use RichEdit control for input/output
  - RTF format preserves charset and codepage info
- Cope with multiple keyboard layouts
  - Respond to WM_INPUTLANGCHANGEREQUEST
- Avoid confusing font lists
  - Use standard font-picker dialog (comdlg32.dll)
  - Use EnumFontFamiliesEx() to filter out CHARSETS

Other opportunities

- “Developing multilanguage applications for Windows NT” (slides)

Summary

- Simplicity is fundamental to your success
- Avoiding top 20 UI mistakes helps you to:
  - Keep your existing customers
  - Expand your reach
- Three things to remember

  1) Deliver great setup and initial experience

  2) Be careful using the file system!

  3) Avoid GUI

PDC User Experience

Must do!

- ***Leveraging Windows 98 and Windows NT 5.0,*** **Wed 4:30 - 5:45**
- ***Top Twenty UI Mistakes in Today’s Apps,*** **Thu 10:00 - 11:15**

Questions And Answers

http://msdn.microsoft.com/developer/userexperience/